# Logique
## TD n°11

Luc Chabassier
chabassier@lsv.fr
Amélie Ledein
ledein@lsv.fr

**Exercise 1 : Resolution**

The signature is $\mathcal{P} = \{ = (2) \}$, $\mathcal{F} = \{ a(0), b(0), f(1) \}$. The goal of the exercise is to use resolution to prove that every injective function over a set of less than two elements is surjective.

We consider the sequent $A_1, A_2, A_3, A_4, A_5 \vdash B$ where:

$$\begin{array}{rcl} A_1 & = & \forall x.\ \forall y.\ (x = y \Rightarrow y = x) \\ A_2 & = & \forall x.\ \forall y.\ \forall z.\ [(x = y \wedge y = z) \Rightarrow x = z] \\ A_3 & = & \forall x.\ \forall y.\ (f(x) = f(y) \Rightarrow x = y) \\ A_4 & = & \forall x. (x = a \vee x = b) \\ B & = & \forall y.\ \exists x.\ (f(x) = y) \end{array}$$

If you have trouble with this exercise, try beginning with Exercise 3 of TD10.

1. Compute the clausal forms $C_1, \ldots, C_4, C$ associated to formulas $A_1, \ldots, A_4, \neg B$. In the following we use the symbol $c$ for the symbol introduced in the skolemization step of the procedure over formula $\neg B$.
2. Derive $C_5 = \neg(f(x) = y) \vee \neg(c = y)$ from clauses $C_1, C_2, C$.
3. Derive $C_6 = \neg(f(f(x)) = y) \vee \neg(f(c) = y)$ from clauses $C_1, C_2, C_3, C$.
4. Derive $C_7 = \neg(f(x) = a) \vee \neg(f(y) = b)$ from clauses $C_4, C_5$.
5. Derive $C_8 = \neg(f(c) = b)$.
6. Derive $C_9 = f(c) = b$.
7. Show that $\perp$ is derivable. Conclude using the correctness of resolution.

**Exercise 2 : Lifting lemma**

Let $T$ be a set of clauses.

Reminder concerning the rules of system $R_0$:

$$\frac{P \vee C \qquad \neg P \vee C'}{C \vee C'}\ \text{IdRes} \qquad \frac{L \vee L \vee C}{L \vee C}\ \text{IdFact} \qquad \frac{C}{C\{x \to t\}}\ \text{Subst}$$

Prove the lifting lemma: if $T'$ is a set of copies of clauses of $T$ and $\theta$ a substitution such that:

- no free variable appears in two different clauses of $T'$,
- $\perp$ is derivable from $\theta T'$ in system $R_0$,

then $\perp$ is derivable from $T$ in $R$.

**Exercise 3 : Logic programming**

Let $\mathcal{F}, \mathcal{P}$ be a signature. We say that a function $h : T(\mathcal{F})^n \longrightarrow T(\mathcal{F})$ is computable by resolution if there exists a set of clauses $C$ written using function symbols in $\mathcal{F}$ and of predicate symbols in $\mathcal{P} \sqcup \{H(n+1)\}$ such that for every $n$-uple of closed terms $t_1, \dots, t_n$ and every closed term $u$ over signature $\mathcal{F}$, $R(u)$ is derivable from $C \cup \{\neg H(t_1, \dots, t_n, x) \vee R(x)\}$ iff $u = h(t_1, \dots, t_n)$ ($R$ is a fresh predicate symbol, i.e. not in $\mathcal{P} \sqcup \{H(n+1)\}$).

1. We work over signature $\mathcal{F} = \{\ 0(0)\ ,\ S(1)\ \}$ and $\mathcal{P} = \emptyset$. We consider predicate symbol $H = Add(3)$ and set of clauses $C$:

$$\begin{array}{l} Add(0, x, x) \\ \neg Add(x, y, z) \quad \vee \quad Add(S(x), y, S(z)) \end{array}$$

   (a) Show that $R(S(S(0)))$ is derivable from set of clause $C$ and clause $\neg Add(S(0), S(0), x) \vee R(x)$.

   (b) We write $S^n(0)$ for the closed term $S(\dots(S(0))\dots)$ with $n$ symbols $S$. Show that addition is computable by resolution using these clauses, where addition is defined over closed terms of $\mathcal{F}$ as $add(S^n(0), S^m(0)) = S^{n+m}(0)$.

2. Working over signature $\mathcal{F} = \{\ 0(0)\ ,\ S(1)\ \}$ and $\mathcal{P} = \{\ Add(3)\ \}$, give the set of clauses computing the multiplication of two natural numbers, where multiplication over closed terms is defined by $mul(S^n(0), S^m(0)) = S^{n \times m}(0)$.
3. Show that the concatenation of lists and reversal of a list are computable by resolution: give the signature and the associated sets of clauses.

There are many other examples: the usual boolean operators, minimun and maximum of two natural numbers, the function $n \mapsto 2^n$...